Tạp chí Tin học và Điều khiển học, T.27, S.1 (2011), 9–22

TỐI ƯU BIỂU THỨC ĐIỀU KIỆN TÌM KIẾM TRONG CÁC TRUY VẤN ĐỐI TƯỢNG SỬ DỤNG CÁC CHỈ MỤC LỒNG

TRƯƠNG NGỌC CHÂU[1], ĐOÀN VĂN BAN[2]

[1] *Trường Đại học Bách khoa, Đại học Đà Nẵng*
[2] *Viện Công nghệ Thông tin*

Tóm tắt. Trong các cơ sở dữ liệu (CSDL) lớn nói chung và CSDL hướng đối tượng (HĐT) nói riêng, các truy vấn trên nó thường chứa biểu thức điều kiện tìm kiếm gồm các phép hội và tuyển phức tạp. Bài báo nghiên cứu và đề xuất cách tối ưu biểu thức điều kiện tìm kiếm phức tạp, trong đó có chứa các biểu thức đường dẫn sử dụng các chỉ mục lồng.

Abstract. In large databases in general and Object-Oriented Database in particular, the queries on databases often contain conditional expression finding including complicated conjunctions and disjunctions. This article proposes optimization technique for complicated finding conditional expression contained path expressions using nested indexes.

1. GIỚI THIỆU

Để tối ưu hóa truy vấn trên CSDL HĐT có rất nhiều hướng tiếp cận khác nhau, nhưng tựu chung có thể chia làm hai mức là mức lôgic và mức vật lý.

Tối ưu hóa ở mức lôgic chủ yếu tập trung vào việc biến đổi ngữ nghĩa nhằm xây dựng các lược đồ thực thi có chi phí thấp nhất có thể. Một vấn đề quan trọng trong bài toán tối ưu truy vấn tổng quát là tối ưu các biểu thức đường dẫn. Trong [4] đã trình bày phương pháp tối ưu các biểu thức đường dẫn chứa trong biểu thức điều kiện tìm kiếm dạng hội hoặc tuyển một cách tổng quát, và phương pháp này không sử dụng bất kỳ chỉ mục nào trên các biểu thức đường dẫn.

Tương tự tối ưu hóa ở mức lôgic, tối ưu ở mức vật lý chủ yếu tập trung vào chi phí lưu trữ và chi phí nhập, xuất đối tượng. Các tác giả trong [2] đã đề xuất giải pháp lưu trữ các đối tượng trên bộ nhớ thứ cấp nhằm nâng cao hiệu suất truy cập.

Một vấn đề khác trong tối ưu truy vấn đối tượng ít được quan tâm là tối ưu các biểu thức điều kiện tìm kiếm phức tạp, trong đó có chứa các biểu thức đường dẫn. Nhóm tác giả Wan-sup Cho [7] đã đề xuất giải pháp tối ưu các biểu thức đường dẫn, gồm các lớp đối tượng dựa trên các chỉ mục đường dẫn trong biểu thức điều kiện tìm kiếm, nhưng chỉ giới hạn đối với biểu thức điều kiện chứa các vị từ nối với nhau bằng phép AND. Một tiếp cận khác của nhóm tác giả Surajit Chaudhuri [5] là đưa ra giải pháp tối ưu biểu thức điều kiện phức tạp trên các truy vấn quan hệ dựa vào lược đồ hợp, giao các chỉ mục. Nhưng phương pháp này chưa xây dựng được một lược đồ xử lý tổng quát cho biểu thức điều kiện bất kỳ.

Từ nhận xét trên, chúng tôi đã nghiên cứu và đề xuất giải pháp tối ưu các biểu thức điều kiện tìm kiếm trong CSDL HĐT một cách tổng quát hơn, trong nó có chứa các biểu thức

đường dẫn với các chỉ mục lồng tương ứng.

Nội dung còn lại của bài báo được tổ chức như sau: Mục 2 trình bày một số khái niệm cơ sở liên quan trong bài báo; Mục 3 mô tả các tham số liên quan đến mô hình chi phí của các thao tác truy cập, hợp và giao chỉ mục lồng; Mục 4 đề xuất giải pháp tối ưu biểu thức điều kiện tìm kiếm phức tạp bằng cách viết lại, phân tích các chi phí truy cập và duyệt chỉ mục, sau đó đề xuất cách tổ chức lại chỉ mục lồng và cuối cùng là kết luận.

2. MỘT SỐ KHÁI NIỆM CƠ SỞ

2.1. Biểu thức đường dẫn và chỉ mục lồng

Trong các điều kiện tìm kiếm của truy vấn trên một lớp có thể được biểu diễn bởi dãy kết hợp của các vị từ lồng có dạng **<biểu thức đường dẫn> <toán tử> <giá trị>**. Một vấn đề quan trọng trong việc hỗ trợ các truy vấn trong các CSDL HĐT là làm cách nào để ước lượng hiệu quả các truy vấn liên quan đến các vị từ lồng.

Định nghĩa 2.1. $o_1.A_1...A_n$ là một biểu thức đường dẫn kết hợp với các lớp $C_1, C_2, ..., C_n$ nếu o_1 là đối tượng thuộc lớp C_1 và A_i là thuộc tính của lớp C_i, mà miền giá trị của A_i có thể là một định danh đối tượng, đối tượng, sưu tập các định danh đối tượng hay đối tượng thuộc lớp C_{i+1}, với $i = 1, 2, ..., n-1$.

Một lớp có thể có nhiều đường dẫn bắt đầu từ nó và lớp này gọi là lớp gốc. Trong bài này chúng tôi chỉ xét biểu thức điều kiện tìm kiếm chứa các biểu thức đường dẫn có cùng lớp gốc.

Định nghĩa 2.2. Cho biểu thức đường dẫn $P = C_1.A_1.A_2.....A_n$, một chỉ mục lồng (NI: Nested Index) trên P được định nghĩa là tập gồm tất cả các bộ (o, s), trong đó: $s = \{o' | \exists(o_1, ..., o_n)$ là một thể hiện của P, với $o' = o_1$ và $o = o_n\}$, phần tử o của bộ (o, s) gọi là khóa (key) của chỉ mục.

Chỉ mục lồng cung cấp một kết hợp trực tiếp giữa một đối tượng cuối và các đối tượng bắt đầu ứng với một đường dẫn.

Ví dụ 2.1 Để thống nhất trong trình bày các ví dụ minh họa, xét CSDL sau:

```
define class NHANSU:
    type tuple( hoTen: String;
          gioiTinh: String;
          namSinh: String;
          )
end NHANSU
define class SINHVIEN inherits NHANSU:
    type tuple( thuocLop: LOP;
          )
end SINHVIEN
```

```
define class KHOA:
    type tuple( tenKhoa: String;
          truongKhoa: NHANSU;
          dsNganh : set(String);
          dsLop: set(LOP);
          )
end KHOA
define class LOP:
    type tuple( tenLop: String;
          chuNhiem: NHANSU;
          banCanSu: set(SINHVIEN);
          khoaHoc: String;
          dsSinhVien: set(SINHVIEN);
          thuocKhoa: KHOA;
          )
end LOP
```

Chỉ mục lồng định nghĩa trên đường dẫn $KHOA.dsLop.dsSinhvien.hoTen$ được kí hiệu là $NI(KHOA.dsLop.dsSinhvien.hoTen)$, là tập các bộ (o, s) gồm hai trường sau:

- khóa chỉ mục o: $SINHVIEN.hoTen$
- con trỏ s: tập các giá trị OID của các đối tượng trong lớp $KHOA$

Nếu các chỉ mục lồng được xây dựng dựa trên các vị từ trong điều kiện tìm kiếm thì việc truy cập đến các lớp liên quan trong CSDL HĐT có thể được thay thế bởi phép duyệt chỉ mục lồng mà thôi. Ví dụ, nếu một chỉ mục lồng được định nghĩa trên đường dẫn: $KHOA.dsLop.dsSinhvien.hoTen$ thì các lớp $LOP, SINHVIEN$ không cần truy cập khi duyệt, để thực hiện các kết nối ẩn giữa chúng. Tuy nhiên, nhược điểm của chỉ mục lồng là cập nhật rất phức tạp và tốn kém [1].

2.2. Mô hình chi phí

Cấu trúc dữ liệu để mô hình hóa cấu trúc chỉ mục lồng là cấu trúc B^+-cây [1].

Các tham số liên quan đến mô hình chi phí

Cho đường dẫn $P = C_1.A_1.A_2.....A_n$, các lớp tương ứng với đường dẫn P là $\{C_1, C_2, ..., C_n\}$. Các tham số sau mô tả các đặt trưng của các lớp và các thuộc tính trong việc mô hình hóa chi phí [1]:

- D_i số các giá trị phân biệt trên thuộc tính A_i, $1 \leq i \leq n$.
- N_i số các đối tượng của lớp C_i, $1 \leq i \leq n$.
- k_i số trung bình các thể hiện của lớp C_i có cùng giá trị trên thuộc tính A_i, $k_i = N_i/D_i$.
- $OIDL$ chiều dài định danh của đối tượng
- PS kích thước trang
- pp chiều dài của một con trỏ trang
- kl chiều dài trung bình của trường $key - value$
- kll kích thước của trường $key - length$
- rl kích thước của trường $record - length$
- $noid$ kích thước của trường no_oids

Chi phí truy cập chỉ mục [1]

Chi phí truy cập và duyệt chỉ mục là chi phí đọc và tìm kiếm danh sách các định danh đối tượng OID tương ứng với vị từ tìm kiếm nào đó. Trong bài này chúng tôi chỉ xét dạng truy vấn với vị từ dạng: $key = value$. Để ước lượng vị từ yêu cầu chỉ cần duyệt một chỉ mục độc lập và có tổng chi phí: *Các trang không phải là nút lá + Các trang nút lá.*

Sau đây là công thức cho biết số các trang chỉ mục được truy cập để ước lượng một vị từ:

$$\begin{array}{ll} A = h + 1 & \text{nếu } X \leq PS \\ A = h + X/PS & \text{nếu } X > PS \end{array}$$

trong đó, h là số các nút không phải là nút lá cần được truy cập (h là chiều cao của B^+-cây), X là kích thước trung bình của một bản ghi chỉ mục nút lá:

$$X = \prod_{i=1}^{n} k_i \times OIDL + kl + ol \quad \text{nếu} \quad X \leq PS$$

$$X = \prod_{i=1}^{n} k_i \times OIDL + kl + ol + DS \quad \text{nếu} \quad X > PS$$

$$\text{với} \quad DS = [\prod_{i=1}^{n} k_i \times OIDL + kl + ol]/PS \times (OIDL + pp)$$

$$ol = kll + rl + noid$$

Chi phí hợp, giao chỉ mục[5]

Hợp và giao 2 danh sách định danh đối tượng OID được thực hiện bằng thao tác băm-trộn (hash-merge). Chi phí của thao tác này được tính một cách tuyến tính theo tổng số các định danh đối tượng OID trong các danh danh được giao hoặc hợp. Ngoài ra giao và hợp chỉ mục còn có thể được thực hiện bởi thao tác sắp xếp-trộn (sort-merge). Thao tác này có chi phí tuyến tính theo tổng số các dòng có liên quan. Ví dụ, nếu hai danh sách định danh đối tượng có chiều dài lần lược là l_1 và l_2 thì chi phí trộn hai danh sách này có thể được xem là $l_1 + l_2$.

3. TỐI ƯU BIỂU THỨC ĐIỀU KIỆN TÌM KIẾM SỬ DỤNG CHỈ MỤC LỒNG

3.1. Đặt vấn đề

Không mất tính tổng quát, chúng ta giả thiết rằng mọi truy vấn đều trả về các định danh (OID) của các đối tượng thỏa mãn điều kiện tìm kiếm. Sau đó, người dùng có thể dựa vào các định danh này để gửi các thông điệp đến các đối tượng và nhận giá trị xác định từ các thuộc tính hoặc gọi các phương thức. Trong bài này chúng tôi giả thiết tất cả các truy vấn đều được ước lượng dựa trên các tệp chỉ mục lồng có sẵn. Khi đó, việc ước lượng sử dụng các kỹ thuật hợp hoặc giao chỉ mục tương ứng với biểu thức điều kiện tìm kiếm gồm các phép hội hoặc tuyển là rất cần thiết. Đặc biệt, kỹ thuật này là tối ưu trong trường hợp sử dụng nhiều hơn một chỉ mục lồng trên các biểu thức đường dẫn có cùng lớp gốc. Kỹ thuật hợp hoặc giao chỉ mục hoạt động như sau: khi vị từ liên quan đến trường khóa của một chỉ mục đang tồn tại, danh sách các OID của các bộ trong tập tin chỉ mục thỏa mãn vị từ được tìm thấy. Khi có một vị từ hội (hay tuyển) nào khác trên các trường khóa của chỉ mục khác, kỹ thuật tạo ra danh sách các OID khác bằng cách sử dụng chỉ mục thứ hai, giao (hay hợp) hai danh sách OID ta có được danh sách các OID thỏa mãn đồng thời hai vị từ đã cho, và tương tự như vậy cho các vị từ còn lại.

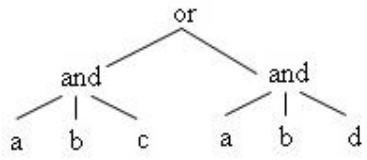
Ví dụ 3.1 Xét biểu thức điều kiện tìm kiếm như sau:

((sv.thuocLop.khoaHoc = '2005-2010') and (sv.thuocLop.banCanSu.gioiTinh $_\forall$ = 'Nữ') and (sv.thuocLop.thuocKhoa.truongKhoa.gioiTinh = 'Nữ')) or
((sv.thuocLop.khoaHoc='2005-2010') AND (sv.thuocLop.banCanSu.gioiTinh $_\forall$ = 'Nữ') and (sv.thuocLop.chuNhiem.namSinh = 1970))

Đặt:

$$a = (sv.thuocLop.khoaHoc = \text{'2005-2010'})$$
$$b = (sv.thuocLop.banCanSu.gioiTinh_{\forall} = \text{'Nữ'})$$
$$c = (sv.thuocLop.thuocKhoa.truongKhoa.gioiTinh = \text{'Nữ'})$$
$$d = (sv.thuocLop.chuNhiem.namSinh = 1970)$$

là các vị từ lồng.



Hình 3.1. Điều kiện E dưới dạng đồ thị

Trong biểu thức trên có 4 đường dẫn xuất hiện trong bốn vị từ tương ứng của biểu thức điều kiện:
$path_1 = sv.thuocLop.khoaHoc$
$path_2 = sv.thuocLop.banCanSu.gioiTinh$
$path_3 = sv.thuocLop.thuocKhoa.truongKhoa.gioiTinh$
$path_4 = sv.thuocLop.chuNhiem.namSinh$
Giả sử cả bốn đường dẫn $path_1$, $path_2$, $path_3$ và $path_4$ đều tồn tại các chỉ mục lồng. Khi đó, lược đồ thực thi biểu thức điều kiện E được cho trong Hình 3.1, trong lược đồ này ta thấy số lần truy cập các chỉ mục lồng $NI(path_1)$ và $NI(path_2)$ lặp lại 2 lần.

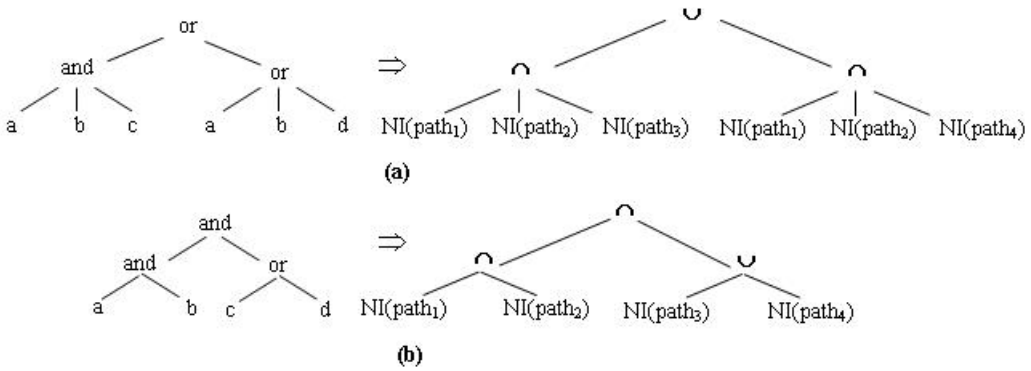
Một vấn đề khác cần được xem xét sau khi biểu thức điều kiện E được viết lại. Giả sử, trong trường hợp các vị từ lồng trong biểu thức điều kiện E chứa các đường dẫn chồng lấp, trong ví dụ này $path_1, path_2$ là hai đường dẫn chồng lấp, giả sử hai chỉ mục lồng được lập cho $path_1, path_2$ tương ứng là $NI(path_1)$ và $NI(path_2)$. Lúc này để ước lượng điều kiện chứa hai vị từ lồng tương đương với việc truy cập và duyệt trên hai chỉ mục lồng $NI(path_1)$ và $NI(path_2)$. Bây giờ nếu ta tiến hành cắt đường dẫn $path_1$ và $path_2$ thành các đường dẫn con và tiến hành lập chỉ mục lồng trên các đường dẫn con này, trong đó có một đường dẫn con là đường dẫn con chung của hai đường dẫn $path_1, path_2$. Lúc này để ước lượng điều kiện chứa hai vị từ lồng tương đương với việc truy cập và duyệt trên các chỉ mục lồng ứng với các đường dẫn con. Vậy, vấn đề đặt ra là: *liệu sau khi cắt thành các đường dẫn con thì chi phí ước lượng có tốt hơn không? và làm cách nào để tiến hành cắt và tổ chức lại chỉ mục lồng sao cho chi phí ước lượng điều kiện là thấp nhất trong trường hợp tổng quát*?

Sau đây chúng tôi đề xuất phương pháp để giải quyết các vấn đề này một cách tối ưu. Giải pháp của chúng tôi đề xuất gồm hai giai đoạn:

- Giai đoạn 1: Viết lại biểu thức điều kiện E bằng cách phân tích E thành các thừa số.
- Giai đoạn 2: Tổ chức lại chỉ mục đối với các đường dẫn chồng lấp xuất hiện trong các vị từ lồng của E sau khi viết lại.

3.2. Viết lại biểu thức điều kiện

Có nhiều lược đồ thực thi chỉ mục khác nhau có thể cho cùng một biểu thức điều kiện truy vấn, phụ thuộc vào các dạng viết lại của nó. Biểu thức điều kiện được xem như là một biểu thức đại số Boolean. Do đó, ta có thể sử dụng các phương pháp đại số để thực hiện tối ưu trên nó. Có thể hình dung rằng dạng tốt nhất của biểu thức Boolean là sử dụng ít nhất số các vị từ. Kỹ thuật để cực tiểu tổng số các vị từ trong biểu thức Boolean là *phân tích biểu thức điều kiện thành các thừa số* [6]. Mục đích của phân tích thành thừa số là viết lại biểu thức đã cho sao cho tổng số các vị từ trong biểu thức viết lại là ít nhất.



Hình 3.2. Lược đồ thực thi chỉ mục của các biểu thức điều kiện tương ứng

Ví dụ 3.2 Xét điều kiện tìm kiếm trong Ví dụ 3.1 (giả sử x and y viết thành xy, x or y viết thành $x + y$):

$$E = (a \quad and \quad b \quad and \quad c) \quad or \quad (a \quad and \quad b \quad and \quad d) = abc + abd$$

Ta có, lược đồ thực thi biểu thức E như ở Hình 3.2(a), trong lược đồ này chỉ mục $NI(path_1)$ và $NI(path_2)$ được truy cập lặp lại 2 lần. Để cực tiểu số lần truy cập chỉ mục, biểu thức E có thể được viết lại tương đương thành $ab(c + d)$. Lược đồ thực thi biểu thức viết lại này được cho trong Hình 3.2(b), lược đồ thực thi này tiết kiệm được 2 lần truy cập chỉ mục trên $NI(path_1)$ và $NI(path_2)$.

Tuy nhiên, vấn đề tối ưu các biểu thức điều kiện tìm kiếm trong truy vấn là phức tạp hơn, vì chúng còn có các chi phí khác kết hợp với mỗi vị từ, như chi phí truy cập, hợp và giao chỉ mục. Vì vậy, biểu thức viết lại tốt nhất có thể không nhất thiết phải luôn chứa ít nhất các vị từ lặp lại. Ví dụ, xét biểu thức $ab + ac$, biểu thức này có thể được viết lại thành $a(b + c)$, nếu xét về số lần truy cập chỉ mục trên các vị từ thì biểu thức sau khi viết lại này tiết kiệm được một lần truy cập chỉ mục trên a và thực hiện ít hơn một lần phép giao chỉ mục. Tuy nhiên điều kiện ban đầu có thể tốt hơn nếu kích thước (danh sách các định danh đối tượng OID) trả về bởi $b + c$ lớn hơn trả về bởi a.

Để phân tích biểu thức điều kiện E thành các thừa số, trước tiên phải chuẩn hóa E thành dạng chuẩn tuyển DNF, sau đó viết lại E thành dạng $D.Q + R$, trong đó: $D \cap Q = \phi$ và D được gọi là số chia, Q là thương và R là phần dư.

Vấn đề quan trọng cần giải quyết ở đây là, nếu cho trước một biểu thức điều kiện E thì E có thể được viết lại dưới nhiều dạng khác nhau sau khi phân tích thành các thừa số dựa trên cơ sở các *số chia* D tìm được. Ví dụ, xét biểu thức điều kiện $E = ac + bc + ad$, E có thể

được viết lại thành $E = a(c + d) + bc$ hoặc $E = c(a + b) + ad$. Vậy, trong trường hợp này là chọn dạng viết lại nào? Việc chọn dạng viết lại nào phụ thuộc vào các số chia tìm được và để chọn số chia nào ta dựa vào các tiêu chí sau đây: tiết kiệm được chi phí truy cập, hợp/giao chỉ mục và chi phí lưu trữ kết quả trung gian. Các số chia thỏa mãn mục tiêu này được gọi là *số chia chất lượng.*

Chất lượng số chia cho phép ta dễ dàng chọn lựa giữa các số chia khác nhau và từ đó quyết định dạng viết lại nào được chọn. Để định nghĩa chất lượng số chia, Chaudhuri [5] đã định nghĩa hàm *Gain* để đo chất lượng của số chia dựa trên các chi phí tiết kiệm được.

Không mất tính tổng quát, giả sử ta xét biểu thức điều kiện: $ax_1 + ax_2 + ... + ax_n$, biểu thức này được viết lại thành: $a(x_1 + x_2 + ... + x_n)$, từ biểu thức viết lại này ta có thể đo được chất lượng của số chia a như sau:

Giả sử l_c là số các định danh đối tượng OID tìm thấy được trong tệp chỉ mục thỏa mãn vị từ c, phép hội ax_i được ước lượng bằng cách giao các danh sách định danh OID tìm thấy được ứng với các vị từ x_i và a. Khi đó, chất lượng của số chia a được biểu diễn qua hàm G sau:

$$G = (n - 1)(k + 1)l_a - \sum_{i=1}^{n}(l_{x_i} - l_{ax_i}) \tag{3.1}$$

trong đó: k là tỉ lệ giữa chi phí phép giao các danh sách chỉ mục và chi phí truy cập các bộ trên đĩa; l_a số các định danh đối tượng thỏa mãn vị từ a; l_{ax_i} số các định danh đối tượng thỏa mãn vị từ hội ax_i; $\sum_{i=1}^{n} l_{x_i}$ số các định danh đối tượng thỏa mãn vị từ tuyển $x_1 + x_2 + ... + x_n$.

Số hạng đầu tiên trong công thức (3.1) là chi phí tiết kiệm đạt được từ việc tránh truy cập lặp lại đối với chỉ mục trên a; số hạng thứ hai biểu diễn chi phí phải trả trong việc tăng chi phí hợp chỉ mục và lợi ích của việc giảm số các phép giao chỉ mục.

Hàm G trong công thức (3.1) phản ánh mức độ hiệu quả của số chia a, hay nói cách khác G ước lượng một cách đáng kể tổng chi phí tiết kiệm được, như chi phí truy cập, lưu trữ, hợp và giao chỉ mục. Từ đây, chúng tôi đề xuất thuật toán viết lại tổng quát biểu thức điều kiện E dựa trên chất lượng số chia tìm được (nghĩa là, số chia có hàm G tương ứng đạt cực đại) từ biểu thức điều kiện E cho trước.

Thuật toán 1. Viết lại biểu thức điều kiện E có tổng chi phí thực thi thấp nhất
Input. Biểu thức E ở dạng chuẩn tuyển DNF.
Output. Biểu thức viết lại của E có tổng chi phí thực thi thấp nhất.
Method.

Function Rewrite(E){
(1). D = tập tất cả các số chia trong E;
(2). If $D = \phi$ Then Return E;
(3). Tìm số chia D_i trong D có hàm G tương ứng đạt cực đại.
(4). Viết lại E thành dạng $D_i.Q + R$;
(5). Return $D_i.Rewrite(Q) + Rewrite(R)$;
}

Định lý 3.1. *Thuật toán 1 dừng và đúng đắn, nghĩa là nó trả về biểu thức viết lại có tổng chi phí thấp nhất.*

Chứng minh.
1. Chứng minh tính dừng: thuật toán dừng khi E không còn số chia.

Hai nhóm tác giả R. K. Brayton [3] và Tsutomu Sasao [6] đã đề xuất một số phương pháp heuristic để tìm số chia D_i, số chia D_i có thể được tìm từ:

- Một vị từ bất kỳ lặp lại ít nhất ở trong hai khối của E.
- Biểu thức chứa các vị từ lặp lại.

Ta thấy sau mỗi bước lặp E được viết lại thành $D.Q + R$, sau đó Q, R được gọi đệ quy để tiếp tục phân tích chúng thành các thừa số. Vì vậy, thuật toán luôn kết thúc khi biểu thức E không có bất kỳ vị từ lặp lại ít nhất ở trong hai khối của E và không chứa biểu thức gồm các vị từ lặp lại.

2. Chứng minh tính đúng đắn của thuật toán: ta cần chứng minh thuật toán trả về biểu thức viết lại có tổng chi phí thấp nhất.

Thật vậy, giả sử chi phí thực thi của biểu thức điều kiện F được ký hiệu là $F.cost$, $soChiaChatLuongNhat$ là số chia trong D tìm được trong bước (3) (có hàm $Gain$ tương ứng đạt cực đại) và luôn thỏa mãn bất đẳng thức:

$$[soChiaChatLuongNhat.Q + R].cost \leq [D_i.Q' + R'].cost \leq E.cost, \forall D_i \subset D$$

mặt khác, sau mỗi bước lặp ta luôn có:

$$[soChiaChatLuongNhat.Rewrite(Q) + Rewrite(R)].cost \leq [soChiaChatLuongNhat.Q + R].cost$$

Vì vậy, lược đồ thực thi dựa trên biểu thức viết lại cuối cùng sẽ có chi phí thấp nhất ■

Ví dụ 3.3 Cho biểu thức điều kiện $E = ace + ade + bce + bde + cf + df$. Tìm dạng viết lại của E.

- Ta có tập các số chia: $D = \{a + b, c + d\}$
- Nếu viết lại E sử dụng số chia $a + b$ ta được $(a + b)(c + d)e + (c + d)f$, còn nếu sử dụng số chia $c + d$ ta được $(c + d)((a + b)e + f)$. Nếu chọn số chia $c + d$ ta tiết kiệm được 10 lần truy cập chỉ mục, trong khi với biểu thức điều kiện E ban đầu phải thực hiện 16 lần truy cập chỉ mục.

3.3. Tổ chức lại chỉ mục

Sau khi biểu thức điều kiện E được tối ưu bằng cách viết lại, chúng tôi tiếp tục phân tích chi phí ước lượng trên các chỉ mục lồng của các đường dẫn trong biểu thức viết lại, đặt biệt là các đường dẫn chồng lấp, sau đó đề xuất giải pháp tổ chức lại chỉ mục lồng sao cho hiệu quả hơn.

Xét hai đường dẫn $P_1 = C_1.A_1.A_2.....A_n$ và $P_2 = C_1.A'_1.A'_2.....A'_m$ ($n \geq 2$ và $m \geq 2$), cả hai đường dẫn có cùng lớp gốc là C_1. Giả sử các lớp kết hợp với chúng là:

$$class(P_1) = \{C_1, C_2, ..., C_n\}, class(P_2) = \{C'_1, C'_2, ..., C'_m\}.$$

Chúng ta xét truy vấn với hai vị từ: $pred_n$ trên thuộc tính lồng A_n và $pred_m$ trên thuộc tính lồng A'_m. Bây giờ chúng tôi tiến hành phân tích và tổ chức lại các chỉ mục lồng trên hai đường dẫn P_1 và P_2 sao cho cực tiểu được chi phí truy cập và duyệt trên nó trong trường hợp P_1, P_2 chồng lấp.

Định nghĩa 3.1. Cho hai biểu thức đường dẫn $P_1 = C_1.A_1...A_n$ và $P_2 = C_1.A'_1...A'_m$, hai đường dẫn P_1 và P_2 được gọi là chồng lấp nếu tồn tại $j < min(n, m)$ sao cho: $A_i = A'_i (1 \leq i \leq j)$ và $A_i \neq A'_i$ nếu $i > j$.

Ví dụ, hai biểu thức đường dẫn:

$SINHVIEN.thuocLop.khoaHoc$ và $SINHVIEN.thuocLop.banCanSu.gioiTinh$ là hai đường dẫn chồng lấp và có $j = 1$.

Trong các trường hợp trước chúng ta xét các đường dẫn với các chỉ mục lồng tương ứng được xác định trên toàn bộ đường dẫn. Tuy nhiên, với một số truy vấn, nó có thể được ước lượng với chi phí hiệu quả hơn nếu cắt đường dẫn thành nhiều đường dẫn con với các chỉ mục lồng được lập trên các đường dẫn con này. Ví dụ, cho đường dẫn $P = SINHVIEN.thuocLop.thuocKhoa.truongKhoa.gioiTinh$ có thể được cắt thành hai đường dẫn con như sau:

$$P^1 = SINHVIEN.thuocLop.thuocKhoa \text{ và } P^2 = KHOA.truongKhoa.gioiTinh.$$

Ngữ nghĩa của hai đường dẫn này như sau: P^1 ứng với mỗi khoa tìm được có tập các sinh viên tương ứng thuộc khoa này; P^2 ứng với mỗi giá trị của *gioitinh* sẽ có tập các khoa có trưởng khoa có giới tính tìm được tương ứng. Trong trường hợp này nếu câu truy vấn có dạng:*Liệt kê danh sách sinh viên thuộc khoa có trưởng khoa là nữ*, sẽ được xử lý như sau:

- Chỉ mục lồng được định nghĩa trên P^2 được truy cập để xác định tất cả các khoa có trưởng khoa là nữ.

- Chỉ mục trên P^1 được truy cập để xác định tất cả các sinh viên thuộc về khoa nào đó được trả về bởi chỉ mục tìm kiếm trên P^2.

Trong trường hợp tổng quát, xét hai đường dẫn có chồng lấp $P_1 = C_1.A_1.A_2.....A_j.A_{j+1}.....A_n$ và $P_2 = C_1.A_1.A_2.....A_j.A'_{j+1}.....A'_m$. Giả sử P_1 được cắt thành $P_1^1 = C_1.A_1.A_2.....A_j$, $P_1^2 = C_{j+1}.A_{j+1}.....A_n$ và P_2 được cắt thành $P_2^1 = C_1.A_1.A_2.....A_j$, $P_2^2 = C_{j+1}.A'_{j+1}.....A'_m$. Khi đó, ta có $P_1^1 = P_2^1$ và P_1^2, P_2^2 là hai đường dẫn không chồng lấp, C_{j+1} được gọi là lớp cắt.

Không mất tính tổng quát, chúng tôi giả thiết rằng chi phí duyệt chỉ mục bằng với chi phí truy cập các bản ghi mút lá của tập tin chỉ mục. Với chỉ mục lồng $NI(P)$ của đường dẫn $P = C_1.A_1.A_2.....A_n$, giả sử giá trị của các tham số kl, ol là không đáng kể so với kích thước của bản ghi chỉ mục X. Ta có, chi phí duyệt chỉ mục lồng tăng khi kích thước bản ghi nút lá tăng và lớn hơn kích thước của trang đĩa. Khi kích thước bản ghi nhỏ hơn kích thước trang đĩa thì chi phí truy cập là hằng số. Vì vậy, ta chỉ xét chi phí duyệt chỉ mục $NI(P)$ khi kích thước bản ghi lớn hơn kích thước trang đĩa. Nhóm tác giả E.Bertino và W.Kim [1] chỉ ra rằng, chỉ mục lồng có thời gian truy cập là hằng nếu $\prod_{i=1}^{n} k_i \leq 500$, ngược lại thời gian truy cập là: $\lceil h + \prod_{i=1}^{n} k_i/500 \rceil$. Vậy, để đơn giản và không mất đi tính tổng quát chúng ta xét chi phí truy cập và duyệt chỉ mục lồng là:

$$NIA(P) = h + \prod_{i=1}^{n} k_i/500 \tag{3.2}$$

Bây giờ chúng ta xét tất cả các trường hợp có thể xảy ra khi ước lượng các truy vấn trên P_1 và P_2.

- *Trường hợp 1*: Các đường dẫn P_1 và P_2 không được cắt. Khi đó, việc ước lượng truy vấn được thực hiện trên trên hai tệp chỉ mục lồng tương ứng của P_1 và P_2. Trong trường hợp này cả hai vị từ lồng được ước lượng bằng cách sử dụng các tệp chỉ mục và sau đó giao/hợp hai kết quả tìm kiếm từ hai chỉ mục tương ứng này lại. Chi phí ước lượng trong trường hợp này là:

$$cost([NI(P_1), NI(P_2)]) = NIA(P_1) + NIA(P_2) \tag{3.3}$$

- *Trường hợp 2*: Đường dẫn P_1 không được cắt, đường dẫn P_2 được cắt thành: $P_2^1 = C_1.A_1.A_2.....A_j$ và $P_2^2 = C_{j+1}.A'_{j+1}.....A'_m$. Chiến lược này tương tự như trường hợp 1, chỉ khác là việc ước lượng vị từ trên thuộc tính lồng A'_m yêu cầu sử dụng hai chỉ mục. Chi phí ước lượng trong trường hợp này là:

$$cost([NI(P_1), NI(P_2^1), NI(P_2^2)]) = NIA(P_1) + NIA(P_2^2) + card(P_2^2) \times NIA(P_2^1) \quad (3.4)$$

trong đó:

$$card(P_2^2) = \prod_{i=j+1}^{m} k'_i$$

là số các thể hiện của lớp C_{j+1} được trả về từ việc tìm kiếm trên chỉ mục lồng của đường dẫn P_2^2.

- *Trường hợp 3*: Các đường dẫn P_1 và P_2 đều được cắt thành các đường dẫn con. Trong trường hợp này các chỉ mục lồng được lập trên các đường dẫn con sau:

$$P_1^1 = C_1.A_1.A_2.....A_j,\ P_1^2 = C_{j+1}.A_{j+1}.....A_n \text{ và } P_2^2 = C_{j+1}.A'_{j+1}.....A'_m.$$

Việc ước lượng truy vấn bao gồm các bước sau:

Bước 1: vị từ lồng trên thuộc tính A'_m được ước lượng đối với lớp cắt C_{j+1} bằng cách sử dụng chỉ mục lồng được sử dụng trên đường dẫn P_2^2. Tương tự, vị từ lồng trên A_n được ước lượng đối với lớp cắt C_{j+1} bằng cách sử dụng chỉ mục lồng được định nghĩa trên P_1^2.

Bước 2: Các kết quả của hai tìm kiếm chỉ mục trong Bước 1 được giao (hợp), tạo ra tập kết quả *Result* gồm các đối tượng (OID) của C_{j+1} thỏa mãn điều kiện của truy vấn. Ứng với mỗi đối tượng trong *Result* một tìm kiếm được thực hiện trên chỉ mục lồng đã định nghĩa trên đường dẫn P_1^1. Các tìm kiếm chỉ mục xác định các đối tượng của lớp C_1 có cùng giá trị với thuộc tính lồng A_j trên các phần tử của tập *Result*. Tập các đối tượng của lớp C_1 đã trả về bởi các tìm kiếm chỉ mục là tập các đối tượng cuối cùng thỏa mãn truy vấn.

Chi phí ước lượng trong trường hợp này là:

$$cost([NI(P_1^1), NI(P_1^2), NI(P_2^2)]) = NIA(P_1^2) + NIA(P_2^2) + card(C_{j+1}) \times NIA(P_1^1) \quad (3.5)$$

trong đó: $card(C_{j+1})$ là số các đối tượng của lớp C_{j+1} thỏa mãn đồng thời hai vị từ lồng $pred_n$ và $pred_m$ nếu hai vị từ nằm trong phép hội và thỏa mãn hoặc vị từ lồng $pred_n$ hoặc $pred_m$ nếu hai vị từ nằm trong phép tuyển.

Vậy, vấn đề đặt ra trong trường hợp này là: *khi nào thì cắt một đường dẫn ban đầu thành nhiều đường dẫn con rồi lập chỉ mục lồng tương ứng trên nó.* Vì đôi khi việc cắt đường dẫn ban đầu thành các đường dẫn con có thể có chi phí ước lượng không tốt hơn ban đầu. Điều này được khẳng định qua các Định lý 3.2, Định lý 3.3 và Định lý 3.4.

Định lý 3.2. *Cho hai đường dẫn chồng lấp P_1 và P_2. Ta luôn có:*

$$cost([NI(P_1), NI(P_2^1), NI(P_2^2)]) > cost([NI(P_1), NI(P_2)])$$

Chứng minh. Từ các công thức tính chi phí (3.3), (3.4) cho trong các trường hợp 1 và trường hợp 2, ta có:

$$\begin{aligned} card(P_2^2) \times NIA(P_2^1) &= \prod_{i=j+1}^{m} k_i' \times (h + \prod_{i=1}^{j} k_i/500) \\ &= h \prod_{i=j+1}^{m} k_i' + \prod_{i=1}^{j} k_i \prod_{i=j+1}^{m} k_i'/500 \\ &= h \prod_{i=j+1}^{m} k_i' + \prod_{i=1}^{m} k_i'/500 \\ NIA(P_2) &= h + \prod_{i=1}^{m} k_i'/500 \end{aligned}$$

Vì $\prod_{i=j+1}^{m} k_i' > 1$ nên $h \prod_{i=j+1}^{m} k_i' > h$, suy ra:

$$card(P_2^2) \times NIA(P_2^1) > NIA(P_2)$$

nên từ đây ta có: $cost([NI(P_1), NI(P_2^1), NI(P_2^2)]) > cost(NI(P_1), NI(P_2))$ ■

Nhận xét: Định lý 3.2 khẳng định rằng nếu đường dẫn P_1 không được cắt thì đường dẫn P_2 cũng không nên được cắt, vì nếu cắt thì chi phí ước lượng sẽ lớn hơn ban đầu.

Định lý 3.3. *Cho hai đường dẫn chồng lấp P_1 và P_2. Giả sử hai đường dẫn này có lớp cắt là C_{j+1}, hai vị từ lồng tương ứng trên P_1 và P_2 nằm trong phép hội, để có:*

$$cost([NI(P_1^1), NI(P_1^2), NI(P_2^2)]) < cost([NI(P_1), NI(P_2)])$$

thì điều kiện sau phải thỏa:

$$\exists \alpha, \beta > 0, \text{ sao cho: } k_1 \times ... \times k_j > (\alpha + \beta + 500\alpha h/D_m')/(\alpha + \beta - \alpha/D_m')$$

Chứng minh. Ta có công thức tính chi phí chi tiết cho từng trường hợp như sau:

$$\begin{aligned} cost([NI(P_1), NI(P_2)]) &= NIA(P_1) + NIA(P_2) \\ &= 2h + \prod_{i=1}^{j} k_i \prod_{i=j+1}^{n} k_i/500 + \prod_{i=1}^{j} k_i \prod_{i=j+1}^{m} k_i'/500 \\ cost([NI(P_1^1), NI(P_1^2), NI(P_2^2)]) &= NIA(P_1^2) + NIA(P_2^2) + card(C_{j+1}) \times NIA(P_1^1) \\ &= 2h + \prod_{i=j+1}^{n} k_i/500 + \prod_{i=j+1}^{m} k_i'/500 \\ &\quad + \prod_{i=j+1}^{n} k_i/D_m' \times (\prod_{i=1}^{j} k_i/500 + h) \end{aligned}$$

trong đó:

$$card(C_{j+1}) = \prod_{i=j+1}^{n} k_i/D_m'$$

(D'_m số các giá trị phân biệt trên thuộc tính lồng A'_m).

Nếu đặt $a = \prod_{i=1}^{j} k_i, b = \prod_{i=j+1}^{n} k_i, c = \prod_{i=j+1}^{m} k'_i$ thì ta thấy luôn

$$\exists \alpha, \beta > 0 : b = \alpha a, c = \beta a.$$

Khi đó:

$$\begin{aligned}
&cost([NI(P_1^1), NI(P_1^2), NI(P_2^2)]) < cost([NI(P_1), NI(P_2)]) \\
&\Leftrightarrow 2h + b/500 + c/500 + b/D'_m(a/500 + h) < 2h + ab/500 + ac/500 \\
&\Leftrightarrow \alpha a/500 + \beta a/500 + \alpha a/D'_m(a/500 + h) < a^2\alpha/500 + a^2\beta/500 \\
&\Leftrightarrow \alpha a/500 + \beta a/500 + \alpha a^2/500D'_m + \alpha ah/D'_m < \alpha a^2/500(\alpha + \beta) \\
&\Leftrightarrow a/500(\alpha + \beta + 500\alpha h/D'_m)/ < a^2/500(\alpha + \beta - \alpha/D'_m) \\
&\Leftrightarrow a > (\alpha + \beta + 500\alpha h/D'_m)/(\alpha + \beta - \alpha/D'_m) \\
&\Leftrightarrow k_1 \times ... \times k_j > (\alpha + \beta + 500\alpha h/D'_m)/(\alpha + \beta - \alpha/D'_m)
\end{aligned}$$

■

Nhận xét: Định lý 3.3 khẳng định rằng các đường dẫn chồng lấp P_1 và P_2 chỉ nên cắt thành các đường dẫn con và lập chỉ mục lồng tương ứng trên nó khi tích các bậc chia sẻ tham chiếu trên đường dẫn con chung là đủ lớn.

Định lý 3.4. *Cho hai đường dẫn chồng lấp P_1 và P_2 và giả sử hai đường dẫn này có lớp cắt là C_{j+1}. Nếu hai vị từ lồng $pred_n$ và $pred_m$ tương ứng trên P_1 và P_2 nằm trong phép tuyển và kết quả tìm kiếm trên hai vị lồng $pred_n$ và $pred_m$ đối với lớp cắt C_{j+1} là không giao nhau thì ta luôn có:*

$$cost([NI(P_1^1), NI(P_1^2), NI(P_2^2)]) > cost([NI(P_1), NI(P_2)])$$

Chứng minh. Ta có công thức tính chi phí chi tiết cho từng trường hợp như sau:

$$\begin{aligned}
cost([NI(P_1), NI(P_2)]) &= NIA(P_1) + NIA(P_2) \\
&= 2h + \prod_{i=1}^{j} k_i \prod_{i=j+1}^{n} k_i/500 + \prod_{i=1}^{j} k_i \prod_{i=j+1}^{m} k'_i/500
\end{aligned}$$

$$\begin{aligned}
cost([NI(P_1^1), NI(P_1^2), NI(P_2^2)]) &= NIA(P_1^2) + NIA(P_2^2) + card(C_{j+1}) \times NIA(P_1^1) \\
&= 2h + \prod_{i=j+1}^{n} k_i/500 + \prod_{i=j+1}^{m} k'_i/500 \\
&\quad + (\prod_{i=j+1}^{n} k_i + \prod_{i=j+1}^{m} k'_i) \times (\prod_{i=1}^{j} k_i/500 + h)
\end{aligned}$$

trong đó:

$$card(C_{j+1}) = \prod_{i=j+1}^{n} k_i + \prod_{i=j+1}^{m} k'_i$$

Đặt $a = \prod_{i=1}^{j} k_i, b = \prod_{i=j+1}^{n} k_i, c = \prod_{i=j+1}^{m} k'_i$. Khi đó, ta có:

$$\begin{aligned} cost([NI(P_1^1), NI(P_1^2), NI(P_2^2)]) &= 2h + b/500 + c/500 + (b+c)(a/500 + h). \\ cost([NI(P_1), NI(P_2)]) &= 2h + ab/500 + ac/500. \end{aligned}$$

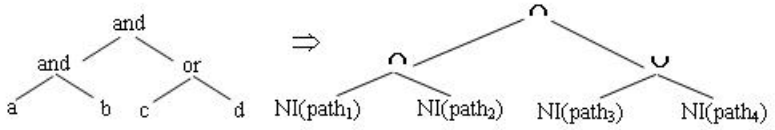
Suy ra:

$cost([NI(P_1^1), NI(P_1^2), NI(P_2^2)]) - cost([NI(P_1), NI(P_2)]) = (b+c)(h + 1/500) > 0$

hay:

$cost([NI(P_1^1), NI(P_1^2), NI(P_2^2)]) > cost([NI(P_1), NI(P_2)])$ ■

Nhận xét: Định lý 3.4 khẳng định rằng các đường dẫn chồng lấp P_1 và P_2 không nên cắt thành các đường dẫn con và lập chỉ mục lồng tương ứng trên chúng, khi kết quả tìm kiếm của các vị từ lồng $pred_n$ và $pred_m$ trong phép hội là không giao nhau.



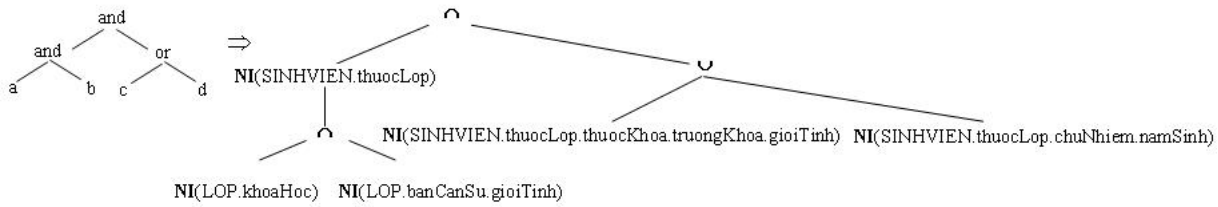
Hình 3.3. Lược đồ thực thi chỉ mục của biểu thức điều kiện E sau khi viết lại

Ví dụ 3.4 Xét biểu thức điều kiện được cho như trong Ví dụ 3.1: $E = abc + abd$. Để tối ưu hóa biểu thức điều kiện này ta lần lược thực hiện theo hai giai đoạn như sau:
- *Giai đoạn 1*: Viết lại biểu thức điều kiện E, ta được $E = ab(c + d)$. Lược đồ thực thi chỉ mục biểu thức điều kiện sau khi viết lại này được cho trong Hình 3.3.
- *Giai đoạn 2*: Xét hai vị từ a, b, giả sử hai vị từ này chứa các đường dẫn $path_1, path_2$ chồng lấp, với $j = 1$ và có lớp cắt là LOP thỏa mãn điều kiện trong Định lý 3.3. Lúc đó lược đồ thực thi được cho như trong Hình 3.4.

Lúc này quá trình thực thi truy vấn được thực hiện qua các bước sau:

Bước 1: Tìm lớp học (OID) học khóa '2005-2010' từ tệp chỉ mục lồng lập trên đường dẫn $LOP.khoaHoc$, tìm lớp học (OID) có ban cán sự lớp là 'nữ' từ tệp chỉ mục lồng lập trên đường dẫn $LOP.banCanSu.gioiTinh$. Giao hai kết quả tìm được và đưa vào $Result$

Bước 2: Ứng với mỗi đối tượng lớp trong $Result$, tiến hành tìm kiếm trên chỉ mục được lập bởi đường dẫn $SINHVIEN.thuocLop$ để xác định các đối tượng sinh viên có lớp tương ứng trong $Result$.



Hình 3.4: Lược đồ thực thi chỉ mục của biểu thức điều kiện E sau khi được tổ chức lại chỉ mục

4. KẾT LUẬN

Bài báo đã tập trung nghiên cứu và đề xuất phương pháp tối ưu các biểu thức điều kiện tìm kiếm gồm các vị từ lồng phức tạp trong các truy vấn CSDL HĐT. Ý tưởng chính của phương pháp là dựa trên kỹ thuật giao, hợp của các chỉ mục, chúng tôi đã đề xuất thuật toán nhằm mục đích viết lại biểu thức điều kiện tìm kiếm phức tạp về dạng đơn giản và có chi phí thực thi thấp hơn. Ngoài ra, chúng tôi cũng đã tiến hành phân tích và đề xuất cách tổ chức lại chỉ mục lồng đối với các đường dẫn chồng lấp xuất hiện trong các vị từ của biểu thức điều kiện sau khi viết lại dựa vào các phân tích chi phí, nhằm giảm thiểu tối đa chi phí truy cập và duyệt chỉ mục.

Ưu điểm của phương pháp đã đề xuất trong bài là cho phép tối ưu các biểu thức điều kiện tìm kiếm bất kỳ bằng cách tạo ra các lược đồ thực thi có các phép hợp và giao chỉ mục, khác với kỹ thuật của nhóm tác giả Wan-sup Cho [7] chỉ tối ưu các biểu thức điều kiện tìm kiếm ở dạng hội, nghĩa là biểu thức điều kiện chỉ chứa phép toán AND.

TÀI LIỆU THAM KHẢO

[1] E.Bertino and W.Kim, Indexing Techniques for Queries on Nested Objects, *IEEE Trans. Knowledge and Data Eng.* **1** (2) (1989) 196–214.

[2] Đoàn Văn Ban, Trương Ngọc Châu, Gom cụm các đối tượng trong CSDL HĐT sử dụng ma trận khoảng cách, *Tạp chí Tin học và Điều khiển học* **25** (2) (2009) 178–187.

[3] R. K. Brayton, D. Hachtel, and A. L. Sangiovarmi- Vincentelli, Multilevel Logic Synthesis, *Proceedings of the IEEE* **7**8 (2) (Feb. 1990) 264–300.

[4] Trương Ngọc Châu, Đoàn Văn Ban, Tối ưu hóa các truy vấn đối tượng có chứa các biểu thức đường dẫn, *Tạp chí Tin học và Điều khiển học* **25** (3) (2009) 201– 213.

[5] Surajit Chaudhuri, Prasanna Ganesan, Sunita Sarawagi, Factorizing Complex Predicates in Queries to Exploit Indexes, *ACM SIGMOD Conference*, San Diego, California, USA, 2003 (361–372).

[6] Tsutomu Sasao, *L*ogic synthesis and optimization, Kluwer academic publishers, USA, 1993.

[7] Wan-sup Cho, Seung-sun Lee, Kyu-young Whang, Yong-ik Yoon, Query Optimization Techniques Utilizing Path Indexes in Object-Oriented Database Systems, *Proc. 5th Int'l Conf. on Database Systems for Advanced Applications (DASFAA)*, Melbourne, Australia, Apr. 1997 (21–29).

Nhận bài ngày 13 - 04 - 2010
Nhận lại sau sửa ngày 25 - 08 - 2010